Research Proposal

Tim von Oldenburg

April 4, 2014

1 Knowledge Area and Research Question

Since the first programmable computers and instructions on punch cards, computing has come a long way. Incredible progress has been made in terms of computing power, technologies and usage areas. We have seen new programming languages and paradigms, we have seen abstractions and whole toolchains to simplify the process of developing software.

Modern Integrated Development Environments (IDEs)[1] were pioneered by systems like the Smalltalk-80 class browser that provided novel, domain-specific ways of navigating information (in this case, source code). However, as Bret Victor argues in his iconic talk, not a lot has changed since then; the progress may be described as evolutionary at best, and includes features such as syntax highlighting, autocompletion and contextual documentation (Victor 2013). In terms of code navigation, developers still rely on file structures, symbol lists or keyword search.

I see large opportunities for improvement in how software is written by improving IDEs, especially in the areas of navigating and evaluating source code. Today's navigational patterns rely on syntax and formal data structures, but not on semantics. Even relatively new patterns, such as presented by Deline et al. (2006) and implemented in editors like Sublime Text[2], concentrate on the actual text instead of its meaning. I want to explore how the navigation of source code can be made more relevant and efficient if it follows *semantics*, for example the internal control flow of the executed program. While a file is a linear medium (a set of lines, which in themselves are a set of characters), the actual program is non-linear.

These thoughts are not completely novel, of course. But while approaches like Visual Programming, as propagated by projects like Scratch (Maloney et al. 2010), serve a certain role, most code is still written as text. Light Table, an experimental IDE currently in a beta testing state, explores different features such as inline evaluation of code and spatial arrangement of semantic code blocks (Granger 2013). I want to contribute to those experimental approaches from the field of interaction design using proper research and validated designs, pursuing the question of how software development can be made more efficient through semantic and context-sensitive code editing environments.

[1] IDE: a software integrating a code editor along with compiler, debugger, code browser and/or other tools.
[2] See `http://www.sublimetext.com/`

2 Methods

Before exploring possible design opportunities, I want to create an overview of the status quo of relevant software development tools and processes. A survey and personal interviews will help identify what is currently used and popular, and will maybe even start to uncover possible shortcomings. Besides of that, research in academic literature and contemporary development projects will show the direction that IDEs are headed towards right now. I expect those methods to result in an incomplete library of patterns used in code editing environments.

The gaps in this library will be areas of opportunity to focus my further work on. I will pick one or more of those areas (depending on the scope of work) and design solutions for them. If possible, and feasible in a timely manner, those designs will be created as high-level prototypes extending a real IDE (possibly either Github's Atom Editor[3] or the Chrome DevTools[4]).

These prototypes will be handed out to software developers to test. Built-in analytics will collect metrics to serve as quantitative measures, whereas user interviews will serve as qualitative measures of design validation. It is as well possible to test the prototypes in a lab environment.

3 Expected Results & Knowledge Contribution

My aim is to provide new knowledge of design patterns for Integrated Development Environments that lead to more productivity and efficiency for software developers. Although I will focus on a certain domain in software development (web development), I strive to find insights that are applicable to a greater range of development environments and paradigms. In addition, I want to demonstrate the practical use of said design patterns by means of solidly implemented, high-level, usable prototypes.

References

Deline et al. 2006 R. DeLine, M. Czerwinski, B. Meyers, G. Venolia, S. Drucker, and G. Robertson, "Code Thumbnails: Using Spatial Memory to Navigate Source Code," in Proceedings of the Visual Languages and Human-Centric Computing, Washington, DC, USA, 2006, pp. 11–18.

Granger 2012 C. Granger "Light Table — a new IDE concept",http://www.chris-granger.com/2012/04/12/light-table---a-new-ide-concept/

Victor 2013 B. Victor "The Future of Programming",http://worrydream.com/#!/TheFutureOfProgramming

Maloney 2010 J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond "The Scratch Programming Language and Environment", Trans. Comput. Educ. 10, 4, Article 16 (November 2010)

[3] See https://atom.io/

[4] See https://developers.google.com/chrome-developer-tools/